Project Title: Multi-Model, Interaction-Oriented Development of Embedded Software

Contract Number: F33615-00-C-1705

Award End Date: 15 Sep. 2003

PI

Eric Engstrom, (612) 951-7318, eric.engstrom@honeywell.com

Team

Kirk Schloegel, David Oglesby, Hazel Shackleton, Devesh Bhatt

Subcontractors and Collaborators

Subcontractors: None

- Due to redirection of our efforts, we have cancelled our previous subcontract to United Defense Incorporated.

Collaborators: Teknowledge, University of Michigan

- Actively working with Teknowledge to implement dynamic (COM-based) integration between DOME and Rational Rose.
- Integration with other phase 1 developers working on Weapon Systems OEP via standard interface(s)
  - University of Michigan AIRES analysis tool via AIF

Problem Description:

- Improve the state of model-based development tools to handle realistically sized problems with support for multi-aspect modeling and robust tool integration capabilities

Project Objective:

Address issues in multi-view model-based development, including:

- Modeling tool with extensible meta-modeling capabilities
  - Support for cross-cutting functional and non-functional constraints
  - Support for multi-model and multi-view integration (at meta-level)
  - Support for “encouraging” formalized reuse (via *archetypes*)
- Code Generator Composition Technology and Infrastructure
  - Support for composable artifact (code/documentation) generation
  - Support for easily specified interoperability in a multi-tool

Success criteria

- Modeling approach and generation techniques adaptable to multiple problem domains
- Applicability to real problems – e.g. Weapon Systems OEP
- Integration of our tools and techniques into a complete development stream
- Elimination of ourselves as meta-modelers and generator authors

So far?

- Ease of transition to the Weapon Systems OEP
- Complete capture of selected Weapon Systems OEP examples

Domain Modeling Environment (DOME)

- Provides a meta-modeling capability that allows construction of cross-linked multi-aspect models
- Provides a composable code-generator capability that allows rapid construction of generators for code and other artifacts
- Provides multi-model views and generators built on the above capabilities, (have been targeted for Weapon Systems OEP)

➢ Inputs

- Human-entered graphical models
- Import models from Rational Rose using COM or XML exchange formats

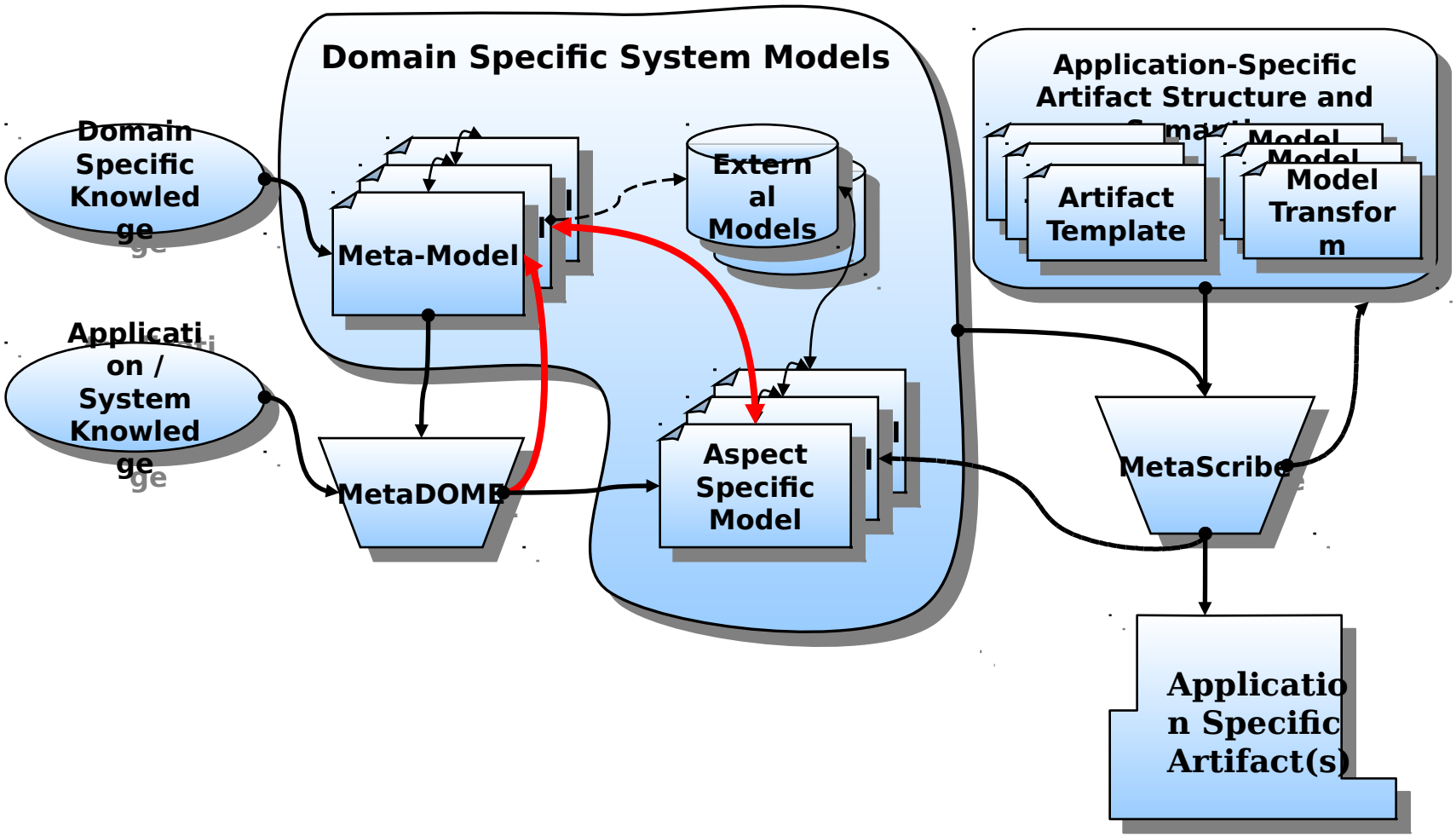
➢ Outputs

- MoBIES Weapon Systems OEP Interchange Formats (AIF and “configuration”)

➢ Interfaces in MoBIES community

- University of Michigan AIRES analysis tools via AIF
- Rational Rose with Teknowledge’s COM interface (work in progress)
- QTIF?

Tool Context in MoBIES



OEP Participation

Weapon Systems OEP

- Role:
  - Provide modeling tool
  - Provide composable code-generator infrastructure
- Midterm Experiments Role:
  - Developed modeling environment and analysis/artifact generators for the OEP planned development scenarios
  - Assisted Boeing in the use of the tool
- Technical POC: Edward Pla – Boeing
- Collaborators: University of Michigan
- Other potential collaborators
  - Vanderbilt: e.g., implementing different model aspects across tools (OITF?)
  - Teknowledge: ROSE Interface via COM (work in progress)
  - CMU: verification tools -- ??

Project Status

Progress since the last PI Meeting:

- Successful midterm experiments on Avionics OEP
  - Integrated event, hardware, and thread modeling views
  - Generation of OEP configuration files and AIF from integrated cross-domain models
  - Modeling-level constraints and analyses
  - Rapid turnaround time for fixes to integration bugs and interchange notation updates
- Increased focus on Weapon Systems OEP
- Transferred technologies to Weapon Systems OEP experiments
  - Modeling support for cross-domain interactions
  - Component-based composable code generation
  - *Archetypes* - a framework for the realization of patterns as first class modeling entities

Weapon Systems OEP Midterm Successes

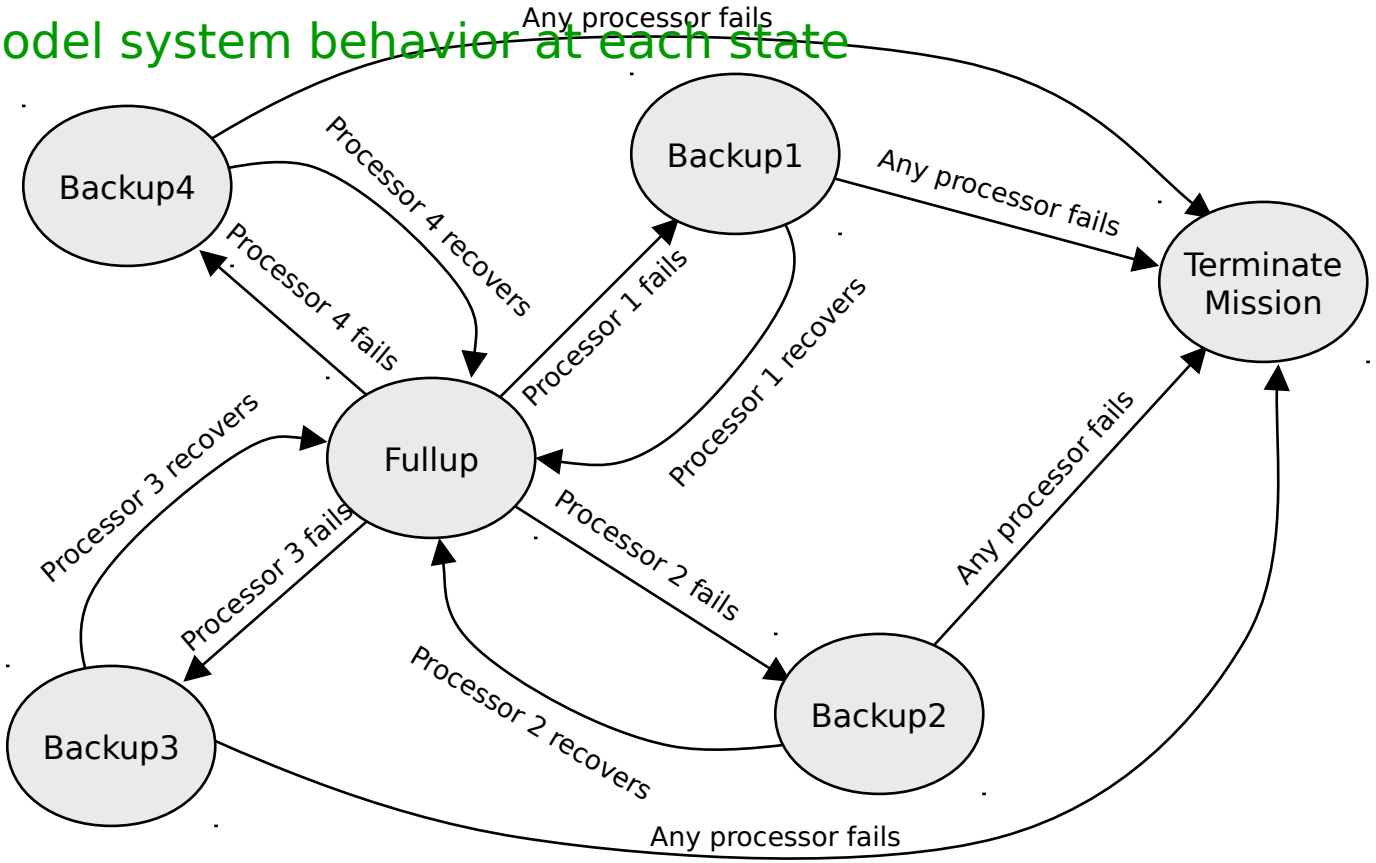
- Rapid transition to Weapon Systems OEP experiments
- Integrated event, hardware, and thread modeling views
- Generation of OEP configuration files and AIF from integrated cross-domain models
- Modeling-level constraints and analyses
  - Event and Facet/Receptacle type checking
  - Model property completeness checking
  - Component-process-hardware mapping
- Rapid turnaround time for fixes to integration bugs and interchange notation updates
  - Our flexible and extensible framework is responsive to these

Progress Since the Midterm

- Interface improvements
  - General usability improvements based upon feedback from users
  - Sorted selection list of model views
  - Error messages in a single non-modal window
  - Faster load times for models
  - Code generation improved (to file)
- Updated code generation in compliance with AIF v1.3
- Parameterized code generation templates
- Working with Teknowledge to implement dynamic integration between DOME and Rational Rose
- Initial work on more advanced modeling issues...
  - Modes
  - Distribution
  - Scalability
  - Reuse

Modeling: fault modes as state-transition diagrams

- Fault mode: degraded system operation (fault tolerance)
  - Model transitions between modes
  - Model system behavior at each state

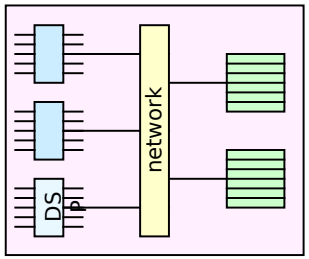


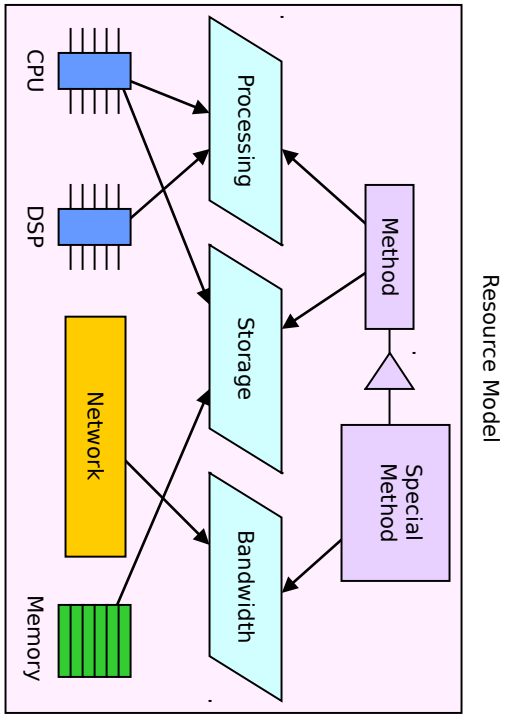
Modeling: Execution modes

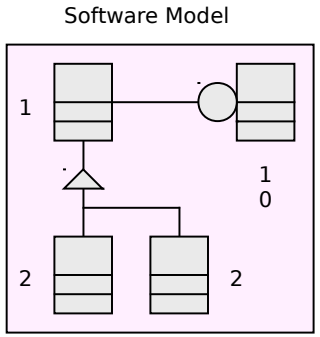
- Execution Modes: Operations optimized for current task
  - Union of “modes” – e.g. Energy, Attack, Steering
- Modeling as state-transition diagrams is promising
  - Structured modeling framework
  - Ensure constraints are maintained (i.e., consistent states are maintained)

Modeling: Distribution

- Distribution is a cross-domain integration problem
  - Assign components to processors, but how to capture interaction?
- We have looked at this problem in the context of resource modeling
  - Akin to how hardware and software aspects interact







Modeling: Scalability

- Multiple views (and files?) of the same model
  - Also, usability by multiple users
- Hundreds of nodes in a model
  - Finding and viewing particular objects of interest
  - Efficient operations such as
    - model loading from file(s)
    - type and consistency checking
    - code generation
    - model exploration issues
      - ***zooming, collapsing models, model mapping***

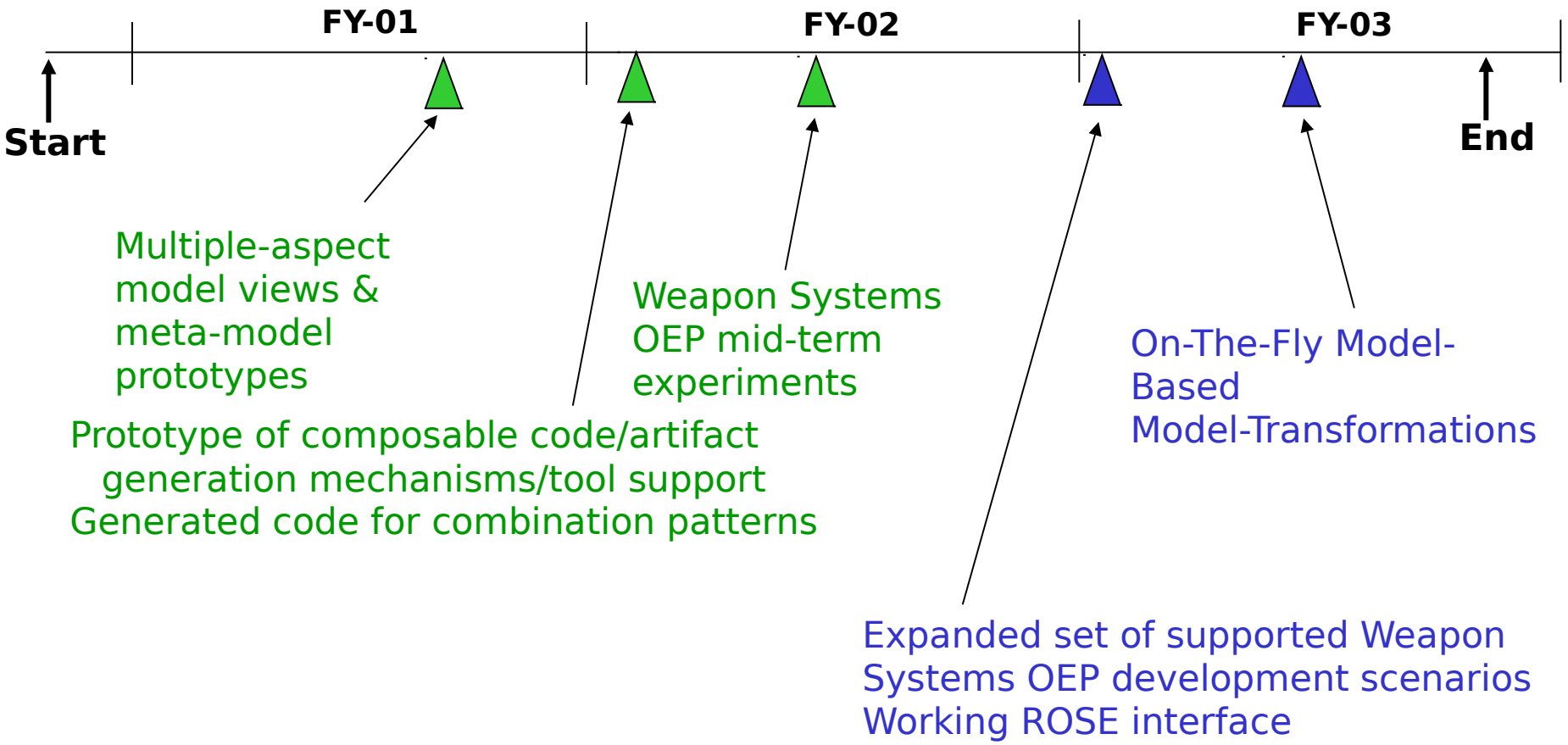
Modeling: Reuse

- Modeling OEP concepts as Archetypes
  - Event channels
    - with Eager and Lazy implementations
  - Components and/or Homes
    - as archetypes and/or implementations?
  - Methods
    - issues of distribution

Near-Term Project Plans

- Plans for next 6 months
  - Continue support of technology experiments using the Weapon Systems OEP
  - Extend composable artifact-generation and tool-integration techniques
  - COM interface to Teknowledge/ROSE bridge
  - Evaluate OTIF and/or HSIF as interfaces with DOME as a modeling tool
- Performance Goals
  - Successfully read in AIF/IIF
  - Working COM-based integration with ROSE
  - Improved usability and scalability of our modeling environment
- Progress and Success Measurement
  - Meet continuing Weapon Systems OEP goals
    - Successfully capture/model/generate more complex examples
    - Round-trip development showing interaction with ROSE

Project Schedule and Milestones



Technology Transition/Transfer

1. Avionics Open Experiment Platform (OEP)

- Participated in mid-term experiments, tool integration

2. Future Combat System (FCS)

- FCS: Honeywell Albuquerque proposed using MoBIES technology as the centerpiece for tools, as part of a network-centric development environment for the upcoming Large Scale Integration (LSI) phase.

3. Honeywell Aerospace Business Technology Platforms

- Platforms provide for the various tools needs for avionics, space systems, and ground vehicles
- MoBIES results are already serving to influence the common next-generation technology; especially tool integration via a common design/architecture model
- Looking at integration of test-generation, test coverage, and generated-artifact verification tools down the road

Program Issues

- Larger focus on the Weapon Systems OEP

- Transferred technologies developed for Crusader
  - Modeling support for cross-domain interactions
  - Component-based composable code generation
  - Archetypes - a framework for the realization of patterns as first class modeling entities